CS534 - Machine Learning

Implementation Assignment 2

Group 50

Sebastian Mueller - 933962290

Derek Helms - 934451909

Vy Bui - 934370552

Instructor: Dr. Xiaoli Fern

The School of Electrical Engineering and Computer Science
Oregon State University

Oct 20, 2022

Part I

Question 1a: what trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best λ value based on the validation accuracy?

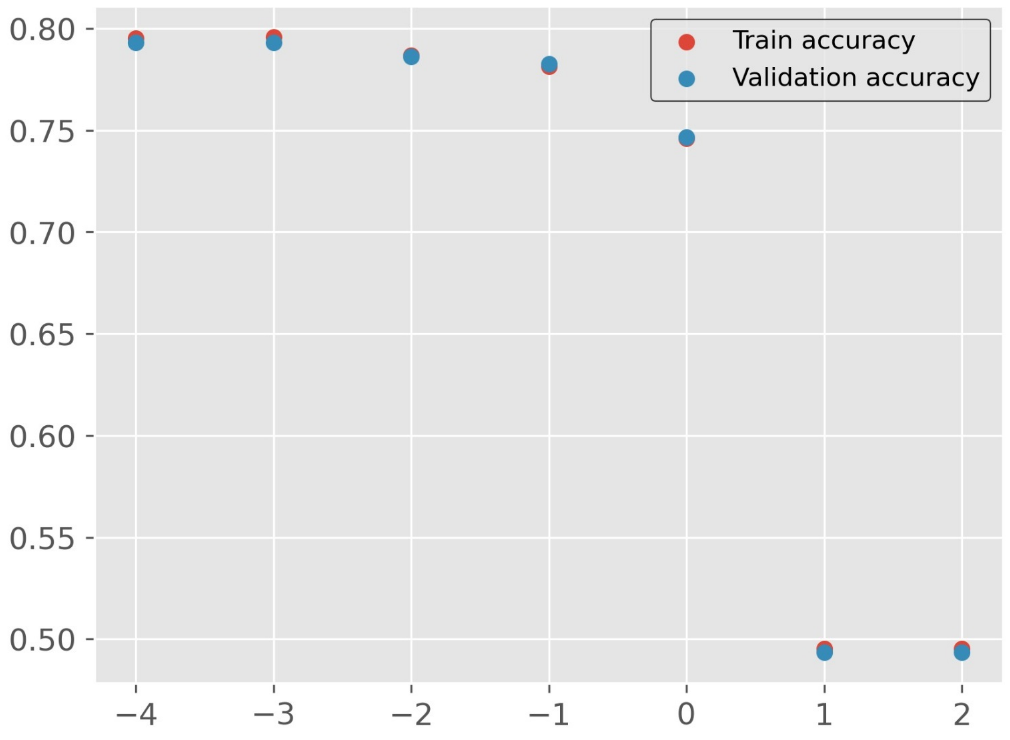

Figure 1: Training and validation accuracy of learned model as function of i used for λ (L2 regularization).

Response:
Regularization is about the bias variance trade off. In general, small λ values are going to over-fit our model to our training data introducing variance, while large λ values will not really allow our model to learn much and introduce bias. This means we should see higher accuracy at small λ vales for our training data, while seeing lower accuracy in our validation data for the same small λ values. We do observe this trend (Figure 1) for $\lambda = 0.0001$ & 0.001. Our best validation was 79.3% at $\lambda = .0001$

With very large λ values (≥ 1) we see accuracy staying around 49% for both training and validation data (Figure 1). This is expected as the large λ value keeps our model from learning much about our training data, so the accuracy of both training and validation data will be poor. Table 2 (see Appendix) provides more information of how our accuracy changed for a given λ as we ran our model.

Trends for increasing λ below value 1 were slightly harder for us to identify. We believe that as you increase λ you should see the train and validation accuracy begin to converge until validation data does better. This is what we do observe as we increase to λ =0, with validation doing better at λ=-1. We believe this is because at more moderate λ values our model is doing a better job at generalizing our data, less influence by training noise, leading to better validation accuracy.

One observation we were confused about was why general accuracy decreased for λ between -4 and -1. We find it hard to believe that $\lambda = .0001$ would not over-fit our model. We do not know how our data was collected or split, so perhaps if training and val come from similar sample set with similar noise, then it is not surprising values are higher for both at smallest λ. Like for example if all the observations for training and validation were taken during an unusually high car accident month. This is something that might not make the data fully independent. Alternatively, it could be explained by the fact that the data may not have very much noise in general. We were wondering if the observed flip of training vs validation accuracy at $\lambda = 0.1$ signifies that the model will do a better job generalizing unseen data, even though over all accuracy is smaller for validation at this λ value.

Question 1b: Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

Response:

features	weights
vehicle damage	2.194210
policy sales channel 26	0.6356899
region code 3	0.633522
policy sales channel 157	0.5664083
policy sales channel 124	0.51777626

Table 1: weights for top five features at $\lambda = .0001$

features	weights
vehicle damage	2.051261
policy sales channel 26	0.5754456
region code 3	0.50653249
policy sales channel 157	0.4723588
policy sales channel 124	0.4678516

Table 2: weights for top five features at $\lambda = .001$

features	weights
vehicle damage	1.495307
policy sales channel 26	0.363763
policy sales channel 124	0.302874
policy sales channel 157	0.219227
region code 28	0.215433

Table 3: weights for top five features at $\lambda = .01$

features	weights
vehicle damage	0.632682
policy sales channel 26	0.129729
vehicle age 0	0.12323
policy sales channel 124	0.123101
region code 28	0.1168539

Table 4: weights for top five features at $\lambda = .1$

yes we do see differences in our top selected features, as we increase λ. As we increase λ the L2 regularizing term drives the weights closer to zero (Tables 1-4). It effectively forces the model to fit while keeping weights as low as possible. With very highest λ top feature weights get very close to zero $\approx .000101$ (table not shown here)

Question 1c:
What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Why?

Response:
We generally see the number of zero features increasing as we increase λ (Figure 3). This is to be expected because as we increase λ the regularization is going to drive more and more weights to zero. There is a big jump with $\lambda = 1$. This must be some kind of threshold that just causes the bias to go very high and push most weights to zero. It follows with the big jump in accuracy we observe in question 1a.

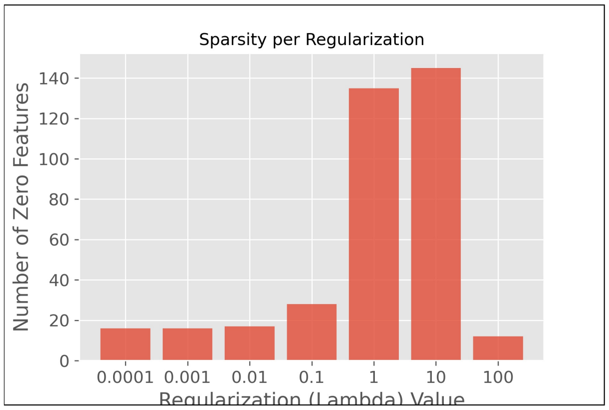

Figure 3: Number of zero weights for different used λ.

One important note is the drop at $\lambda = 100$. This value caused our model to "collapse" and we could not get it to converge, most likely because the big regularizing term keeps the model from getting a good enough fit to pop out from the gradient decent.

Part II

Question 2a:
What are some the key differences do you observe comparing the results obtained using noisy training data to those of part 1? What do you think is the effect of regularization on the model's robustness to noise in the training set? Why?

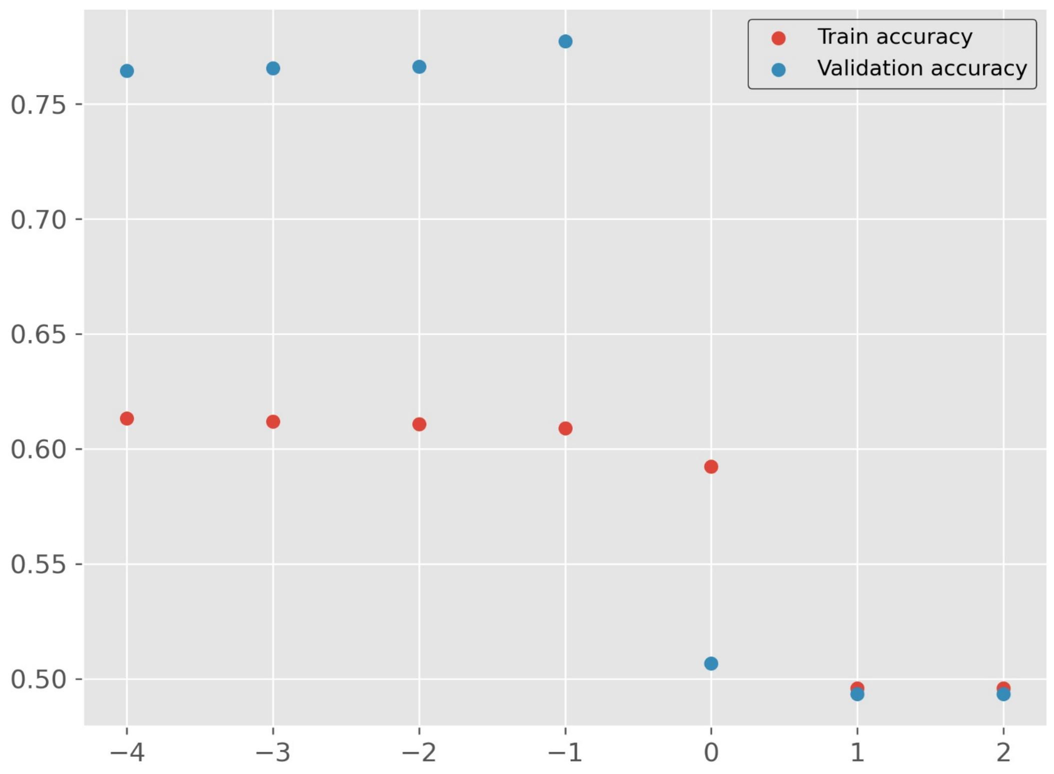

Figure 4: Training and validation accuracy of learned model as function of i used for λ (Noisy, L2 regularization).

Response:
We see a huge difference between training and validation accuracy for this experiment, specifically with validation accuracy being much higher for smaller λ values. The main goal of regularization is to penalize noise, so when introducing noise to the model, it did a good job at "ignoring" it, and allowed us to still produce a well fitting model. This showcases how regularization can significantly increase the robustness of a model.

Of course the model follows the same trend of over overall decreasing accuracy at a certain large value of λ. This is to be expected with or without noise, because large λ will just make the model too biased.

Another interesting difference from part I is in the sparsity plots (Figure 5).

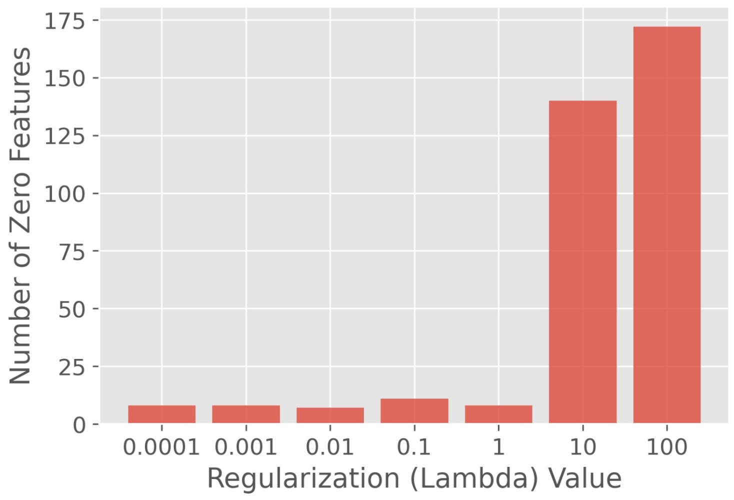

Figure 5: Sparsity of different λ values (L2 regularization, noisy data).

In general, we see less features pushed to zero (excluding large λ's). We were not entirely sure why this was, but we figured maybe because the regularizing term was "competing" harder again the extra added noise, so it was not getting as many terms pushed to zero? This might be the case, because overall this model was slightly less accurate for the validation data at optimal λ than part 1. Which could be caused by features that should be zero not getting there in this model.

Lastly, additional information about the model, like accuracy per iteration is showcased in Figure 6 (see Appendix). Interestingly, you can see how the curves in this figure look more "bumpy", we decided that it must have been from the introduced noise.

Part III

Question 3a:
what trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best λ value based on the validation accuracy?

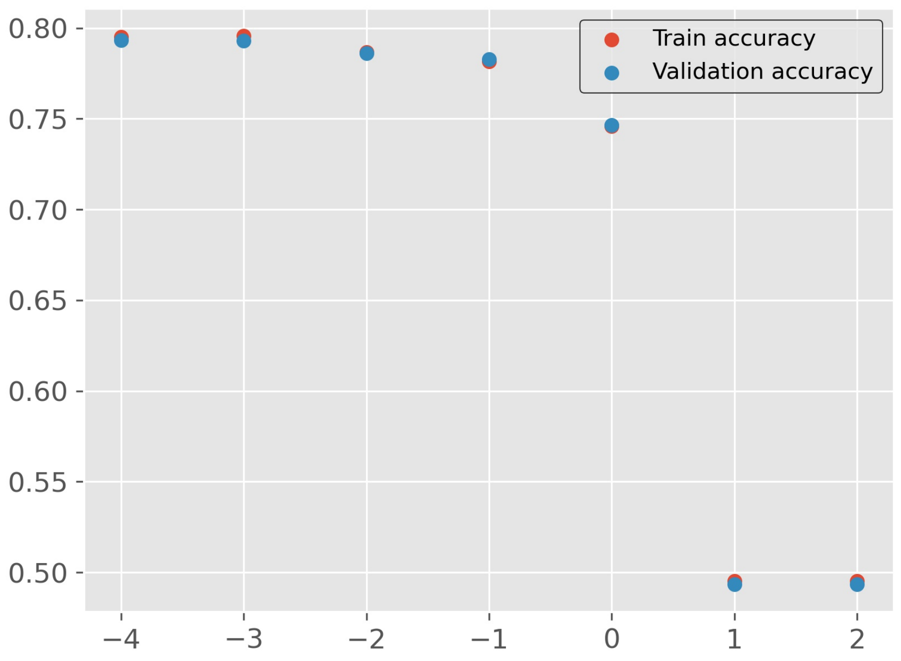

Figure 7: Training and validation accuracy of learned model as function of i used for λ (L2 regularization).

Response:
We do observe general pattern of regularization (see question 1a) happening slightly for L1. In Figure 7 we can see that the red dot for λ = .0001 & .001 achieves slightly higher max accuracy for training vs validation, until λ = .01, where validation is more accurate – this is our best accuracy for validation (78.39% at λ=.01). Since large λ values do not learn much we should see poor accuracy for both training and validation, which we do observe for λ values .1, 1, 10, 100. More information about our accuracy for training and validation data can be seen in Table 7 (see Appendix)

Question 3b:
Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

Response:

features	weights
previously insured	-0.798975
vehicle damage	0.750536
driving license	-0.309737
policy sales channel 124	0.260415
vehicle age 1	-0.256555

Table 5: weights for top five features at λ = .0001

features	weights
Previously insured	-1.257391
vehicle damage	1.144427
policy sales channel 152	-0.393542
driving license	-0.392308
vehicle age 1	-0.382580

Table 6: weights for top five features at λ = .001

features	weights
Previously insured	-0.990252
vehicle damage	0.939625
policy sales channel 152	-0.291322
vehicle age 1	-0.284638
policy sales channel 124	0.169089

Table 7: weights for top five features at λ = .01

features	weights
age	0.109821
vehicle damage	0.094218
vehicle age 2	0.046124
policy sales channel 157	0.045359
policy sales channel 156	0.042267

Table 8: weights for top five features at λ = .1

We observed that increasing λ's generally brought weight value closer to 0. Table 5-8 showcase increasing λ and the corresponding top 5 largest feature weights. λ = .001 gave us our most accurate model. As we increase λ value above our best λ, we can see that absolute value of feature weights are decreasing. Note that these weights are decreasing at different rates; this is showcased when we see a new feature jump rank, like age in Table 7 vs Table 8. Eventually, very high λ's drive the rates to zero. This was the case with $\lambda \geq 1$ (not shown here).

When looking at λ smaller than our best λ we see absolute value of weight also decrease. We are not entirely sure why this is, we initially thought it should increase. We hypothesize that maybe since the model is over-fitting, the weight for "true important" features is in a sense being distributed amongst other features that are actually "not important". The fact that the model is really learning the noise means it may inaccurately assign weights, leading to spread of weight being larger, meaning any one feature may have less weight.

Question 3c:
What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Is this trend different from what you observed in 1(c)? Provide your explanation for your observation.

Response:
We observed a significant jump in sparsity once our model used $\lambda \geq 1$ – with all feature weights eventually going to zero (Figure 8). In general, L1 regularization in a sense "encourages sparsity"

and can drive weights to zero, due to the characteristic sharp edges of the corresponding isomorphic surface falling on an axis. So its not surprising that we see zero weight features for all λ tried. What causes the massive jump to all zero weights is a little less obvious to us. We essentially think it does this because with high λ our model has a lot of bias, thus it considers all weights to be uninformative and drives them to zero. In general however, as we increase λ we expect to see more values pushed to zero.

This is a different trend than what we see in 1c (ignoring bound we consider zero). L2 regularization will usually not push weights to an actual zero. This can be explained by the difference in shape of the isomorphic surfaces. L2 has curved surface while L1 has pointed surface.

One observation we had trouble figuring out was why sparsity had a small increase for λ=0.01, to then decrease in the next λ iteration. Perhaps it was an unusual anomaly or outlier with our data?

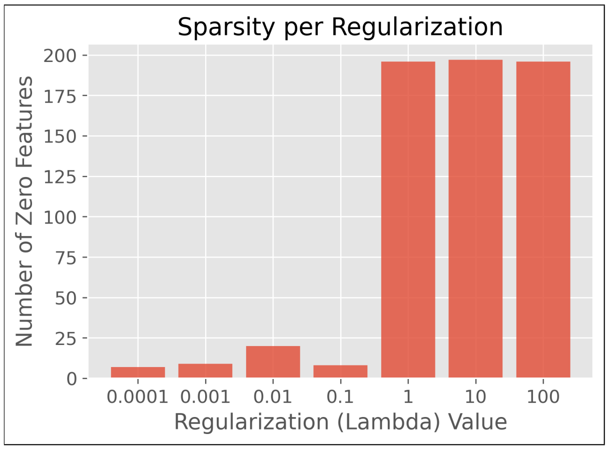

Figure 8: Number of zero weights for different used .

Question 3d:
What are the key differences between the two regularization methods observed on this data set? Specifically, which method achieves the best validation accuracy? Which method is more sensitive to the choice of the regularization parameter for this data? Which method produced sparser feature weights? What are the advantages and disadvantages of each method in general?

Response:

We found the L2 method gave us the best accuracy (79.3%). It was hard for us to discern from our data which regularization method was more sensitive to choice of regularizing parameter. Our intuition says L2 is more sensitive to regularization parameter. We thought this because L2 is better at accounting for outliers in the data, it penalizes more for high value of outlier. This lead us to believe it may be more sensitive to parameter choice. Additionally, we had trouble getting it converge for very large values, which further lead us to believe its more sensitive

We found that L2 produced sparser feature weights. It's important to note if we are solely talking about sparsity being zero weight values then L1 makes a sparser model; however, with the threshold given for L2, L2 seemed to be the more supercity making model.

Another interesting difference we noticed is that the L1 model converged with many less iterations than L2 model, as seen in figures 2 & 5 (see Appendix).

L2 and L1 regularization offer different advantages and disadvantages. For example, L1 is helpful for identify sparsity, which can then help with feature selection. This may go against some of the results shown above; however, L1 forcing values to zero, not just close to zero, will ultimately make it better for understanding sparsity for feature selection. L1 is also more robust to outliers in the data since it is not taking the square value like L2. One obvious advantage we noticed about L2 is that it is more accurate.

Discussion

Many unexpected observations and thoughts were embedded into the actual questions for this project. In general though, we saw a lot about regularization in this project, the most profound for us being how regularization increases model robustness. We also could clearly see how changing the the value of you your regularizing hyper-parameter can really affect the accuracy of your model, with some values very much over fitting or under-fitting. This highlights the care needed for this selection problem.

Appendix

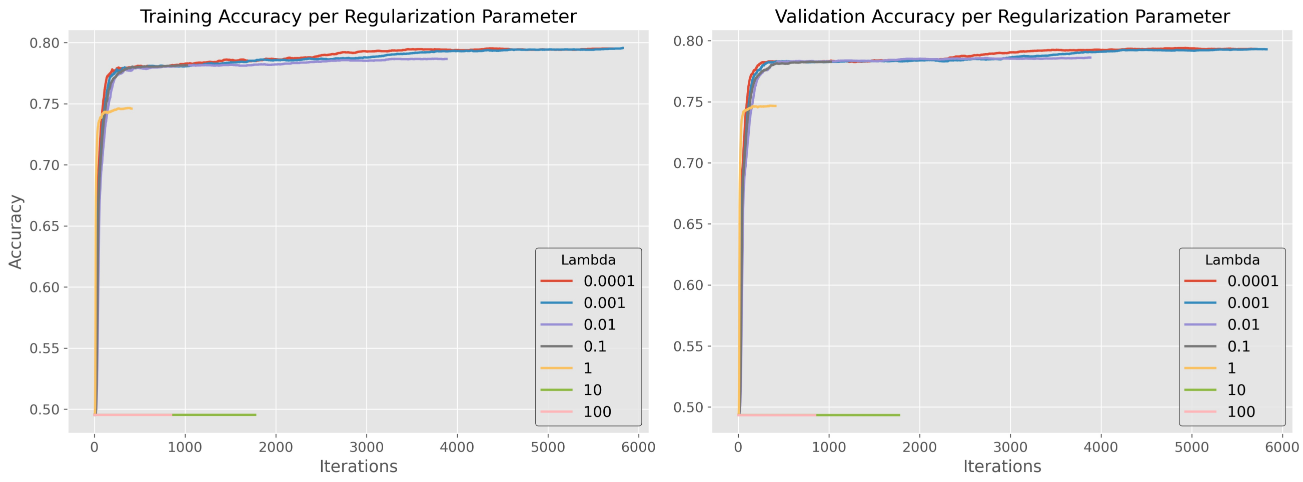

Figure 2: Accuracy of different λ values for training and validation data (L2 regularization).

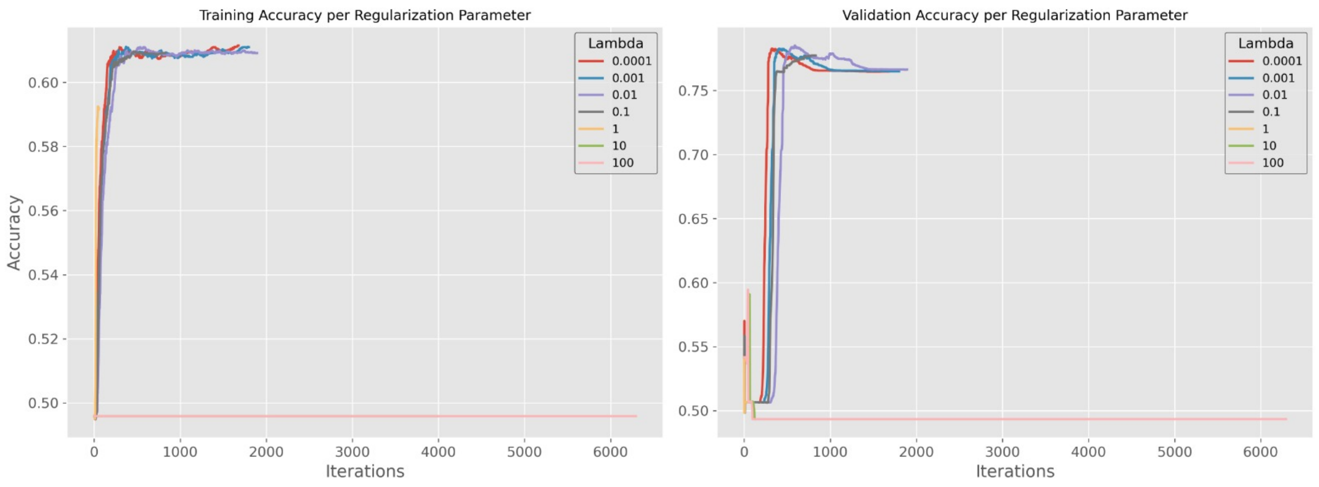

Figure 6: Accuracy of different λ values for training and validation data (L2 regularization, noisy data).

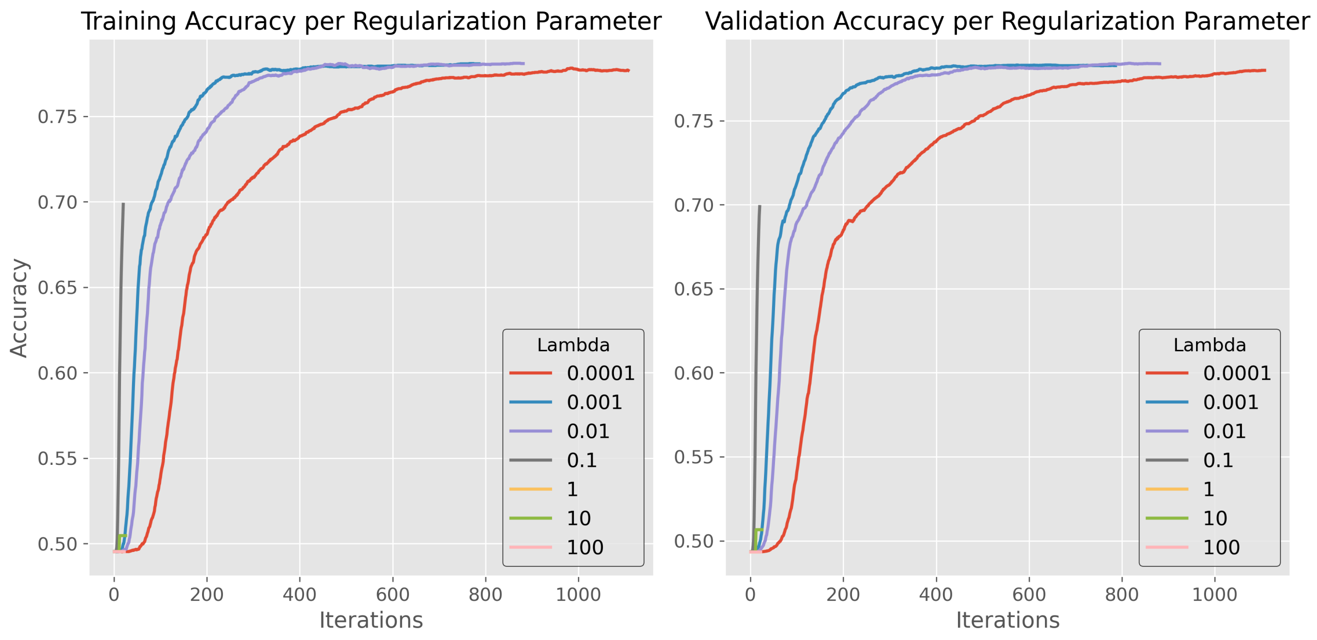

Figure 7: Accuracy of different λ values for training and validation data (L1 regularization).